

Amendments to the Specification:

Please replace paragraph [0047] with the following paragraph.

[0047] Passive element 412 may represent comment 312. Comment 312 and Main Method 310 are included within the class entity 314, represented in the model as graphical entity 414. Namespace Demo 316 may be modeled as 416 as illustrated in FIG. 4-416. A namespace may span multiple files. Similarly a namespace entity may be contained or included within multiple files as Demo 416 is included within the Class1 file 420 so that a many-to-many relationship may exist between files and namespaces.

Please replace paragraph [0012] with the following paragraph.

[0012] FIGs. 6a-6e are exemplary user interfaces in accordance with one aspect of the invention.

Please replace paragraph [0027] with the following paragraph.

[0027] In some embodiments of the invention, the system may include a selector 206 for generating a list of languages 214 in which code is to be generated. List 214 and model 212 may be input to code generator 208. Code generator 208 may generate one or more code modules 212a, 212b, 212c, etc., in the languages specified in list 214. For example, if C++ and C# were selected (and thus contained in list 214), code 212a may be code generated in C++ and code 212b may be code generated in C#. Code generator 208 may generate one or more modules of code in one or more languages, including but not limited to Ada95, Algol, APL, BASIC, C, C#, C++, Clips, COBOL, Common Lisp, Component Pascal, Concurrent Pascal, Delphi®, Eiffel, F#, Forth, FORTRAN, Haskell, Java®, JavaScript®, (ECMAScript), Jess, Joy, Lisp, M, Mercury, ML, Modula-2, NewtonScript, Oberon, Objective C, Ocaml, Occam2, OPS5, OPS-5, Perl, Perl 5, PHP, Pict, Poplog, PostScript, PowerBuilder, Prolog, Python, Q, Rapira, Ruby, Sather, Scheme, ~~Self~~-Self, Simula, Smalltalk, TCL, VBScript, ~~VB Script~~, or VISUAL BASIC®. Any number of languages may be listed in list 214 and any number of modules may be generated.

Please replace paragraph [0041] with the following paragraph.

[0041] The user interface may prompt or allow additional lines of code to be allowed, (e.g., line 304 (“string receiver;”), line 306 (“~~receiver~~generator= 0”), line 308 (“receiver = generator.ToString();”) and so on. For example, now suppose that line 304 is considered. In line 304 (“string receiver;”) a variable “receiver” is declared to be of type “string”. As can be seen from FIG. 4, the graphical representation 404 of line 304 is very similar to the graphical representation of line 302.

Please replace paragraph [0045] with the following paragraph.

[0045] The user interface may then request or allow identification of a second code entity, identification of allowable values for the second code entity (the right hand operand), if the second code entity is an object, if it is an object, what kind of object it is, and if the assignment is a conversion (the type of code entity one is being changed by assignment to code entity two) or a member (code entity one is being assigned the value of a member of code entity two). Suppose, for example, “generator.ToString()” (308c) is entered. Because of the syntax of code entity 308c, (notably in this case, the period 308e and parentheses 308g), the user interface may understand that “generator” 308d is an object, “ToString” 308f a method or function to be performed on object “generator” and the empty parentheses to indicate that no arguments have to be sent to method “ToString” 308f. Alternatively, user interface may receive “generator” 308d as illustrated in exemplary user interface FIG. 6c- and request identification of the type of the right operand 610 (e.g., object, etc.). The type of operation (e.g., cast or member access) may then be requested, as illustrated in FIG. 6d. If, for example, access member 612 is selected, the exemplary user interface FIG. 6e may be displayed, and the list box selections 614 displayed by selecting the down arrow 616. If for example, the type “object” is entered, user interface may request identification of the type of object. In some embodiments of the invention, an object such as exemplary object “generator” 308d is distinguished by displaying object “generator” 408e in a box 408f with method “ToString” 408g in a box 408h within box 408f.